

(51) Int.Cl. ⁷	識別記号	F I	マーク [®] (参考)
G 06 F 13/00	3 5 3	G 06 F 13/00	3 5 3 B 5 B 0 8 5
3/00	6 5 2	3/00	6 5 2 A 5 B 0 8 9
15/00	3 1 0	15/00	3 1 0 A 5 E 5 0 1

審査請求 未請求 請求項の数11 O.L (全 14 頁)

(21)出願番号	特願平10-324044	(71)出願人	000001007 キヤノン株式会社 東京都大田区下丸子3丁目30番2号
(22)出願日	平成10年11月13日(1998.11.13)	(72)発明者	山田 哲也 東京都大田区下丸子3丁目30番2号 キヤ ノン株式会社内
		(74)代理人	100076428 弁理士 大塚 康徳 (外2名)

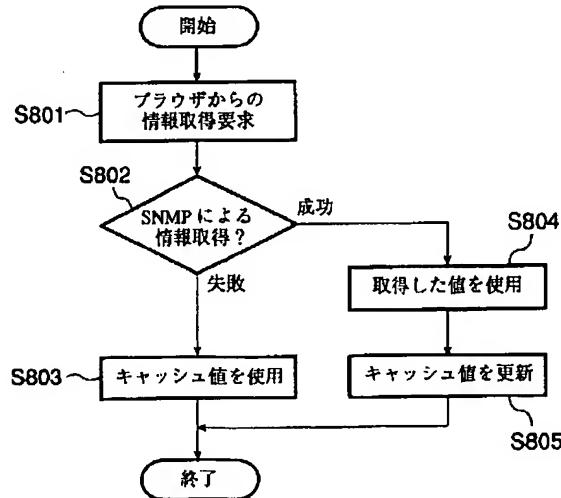
最終頁に続く

(54)【発明の名称】 ネットワークデバイスの情報管理方法及びその装置、記憶媒体

(57)【要約】

【課題】WebブラウザからCGIを経由して外部アプリケーションを起動して、ネットワークワークデバイスの情報を取得するシステムにおいては、アプリケーションがSNMPによる情報取得に失敗した場合、表示する情報がなくなってしまう問題があった。

【解決手段】ブラウザがサーバに対して表示すべき情報の取得要求を出し(S801)、表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行い、情報取得の成否を判定する(S802)。情報取得に成功した場合は、取得値を表示し(S804)、その値をキャッシュ値として更新する(S805)。情報取得に失敗した場合は、キャッシュ値として保存されていた既取得情報を表示する(S803)。



【特許請求の範囲】

【請求項1】 ネットワーク上から所定のインターフェースを経由してサーバのアプリケーションを起動し、取得するネットワークデバイスの情報を管理する方法であつて、

ブラウザがサーバに対して表示すべき情報の取得要求を出すステップと、
前記表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行うステップと、
前記情報取得の成否を判定するステップと、
前記判定に基づき、前記情報取得に成功した場合に、取得した値を表示するステップと、
前記取得した値をキャッシュ値として更新するステップと、

前記判定に基づき、前記情報取得に失敗した場合に、キャッシュ値として保存されていた既取得情報を表示するステップと、

を備えることを特徴とするネットワークデバイスの情報管理方法。

【請求項2】 ネットワーク上から所定のインターフェースを経由してサーバのアプリケーションを起動し、取得するネットワークデバイスの情報を管理する装置であつて、

ブラウザがサーバに対して表示すべき情報の取得要求を出す手段と、
前記表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行う手段と、
前記情報取得の成否を判定する手段と、
前記判定に基づき、前記情報取得に成功した場合に、取得した値を表示する手段と、
前記取得した値をキャッシュ値として更新する手段と、
前記判定に基づき、前記情報取得に失敗した場合に、キャッシュ値として保存されていた既取得情報を表示する手段と、
を備えることを特徴とするネットワークデバイスの情報管理装置。

【請求項3】 ネットワーク上から所定のインターフェースを経由してサーバのアプリケーションを起動し、ネットワークデバイスの情報を取得するプログラムを記憶するコンピュータ可読の記憶媒体であつて、該プログラムが、

ブラウザがサーバに対して表示すべき情報の取得要求を出すステップのコードと、
前記表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行うステップのコードと、

前記情報取得の成否を判定するステップのコードと、
前記判定に基づき、前記情報取得に成功した場合に、取得した値を表示するステップのコードと、
前記取得した値をキャッシュ値として更新するステップ

のコードと、

前記判定に基づき、前記情報取得に失敗した場合に、キャッシュ値として保存されていた既取得情報を表示するステップのコードと、
を備えることを特徴とする記憶媒体。

【請求項4】 前記ネットワークデバイスにはプリンタが含まれることを特徴とする請求項1に記載のネットワークデバイスの情報管理方法。

【請求項5】 前記所定のインターフェースはCommon Gateway Interfaceであることを特徴とする請求項1に記載のネットワークデバイスの情報管理方法。

【請求項6】 前記ネットワークデバイスにはプリンタが含まれることを特徴とする請求項2に記載のネットワークデバイスの情報管理装置。

【請求項7】 前記所定のインターフェースはCommon Gateway Interfaceであることを特徴とする請求項2に記載のネットワークデバイスの情報管理装置。

【請求項8】 前記キャッシュ値として保存されていた既取得情報を表示するステップは、情報取得に成功した場合と区別するための表示制御を行うことを特徴とする請求項1に記載のネットワークデバイスの情報管理方法。

【請求項9】 前記表示制御は文字列の書体若しくは、色彩を変更することにより区別することを特徴とする請求項8に記載のネットワークデバイスの情報管理方法。

【請求項10】 前記キャッシュ値として保存されていた既取得情報を表示する手段は、情報取得に成功した場合と区別するための表示制御を行うことを特徴とする請求項2に記載のネットワークデバイスの情報管理装置。

【請求項11】 前記表示制御は文字列の書体若しくは、色彩を変更することにより区別することを特徴とする請求項10に記載のネットワークデバイスの情報管理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、コンピュータネットワークに関し、具体的には、ネットワーク管理ソフトウェアが管理するネットワークデバイスの情報管理方法及びその方法を実行するための装置、また、その情報管理方法をコンピュータで実行するためのプログラムを記憶したコンピュータ可読の記憶媒体に関する。

【0002】

【従来の技術】 図1は、プリンタをネットワークに接続するためのネットワークボード(NB)101を、開放型アーキテクチャをもつプリンタ102につなげた場合を示す図である。NB101は、ローカルエリアネットワーク(LAN)100に、同軸コネクタをもつEthernetインターフェース10Base-Tや、RJ-45をもつ10Base-TなどのLANインターフェースを介して接続される。

【0003】PC103、PC104などの、複数のパソコンコンピュータ(PC)もまた、LAN100に接続されており、ネットワークオペレーティングシステムの制御のもと、これらのPCはNB101と通信することができる。従って、PCの1つ、例えばPC103を、ネットワーク管理用に使用することができる。また、PCに、PC104に接続されているプリンタ105のようなローカルプリンタを接続してもよい。

【0004】LAN100には、ファイルサーバ106が接続されており、ファイルサーバ106は大容量(例えば、100億バイト)のネットワークディスク107に記憶されたファイルへのアクセスを管理する。

【0005】プリントサーバ108は、接続された複数のプリンタ109、または遠隔地にあるプリンタ105などのプリンタに印刷を行わせる。また、他の図示しない周辺機器をLAN100に接続してもよい。

【0006】また、LAN100には、WWWサーバ150が接続されており、WWWサーバ150上にインストールされたネットワーク管理ソフトウェアが生成したHTMLドキュメントを、PC103上にインストールされたWWWブラウザを使って表示したり、PC103上のWWWブラウザ上で行ったプリンタの設定を、WWWサーバ150上のネットワーク管理ソフトウェアを介して特定のプリンタに送信することもできる。

【0007】更に詳しくは、図1に示すネットワークは、様々なネットワークメンバ間で効率良く通信を行うために、NovellやUNIXのソフトウェアなどのネットワークソフトウェアを使用することができる。どのネットワークソフトウェアを使用することも可能であるが、例えば、Novell社のNetWare(Novell社の登録商標。以下、省略)ソフトウェアを使用することができる。このソフトウェアパッケージに関する詳細な説明は、NetWareパッケージに同梱されているオンラインマニュアルを参照のこと。これは、Novell社からNetWareパッケージと共に購入することができる。

【0008】簡潔に説明すると、ファイルサーバ106は、LANメンバ間でファイルの受信や、記憶、キューイング、キャッシング、および送信を行う、ファイル管理部としての役割を果たす。例えば、PC103およびPC104のそれぞれにより作成されたデータファイルは、ファイルサーバ106へ送られ、ファイルサーバ106は、これらのデータファイルを順に並べ、そしてプリントサーバ108からのコマンドに従って、並べられたデータファイルをプリンタ109の1つに送信する。

【0009】PC103とPC104は、それぞれデータファイルの生成や、生成したデータファイルのLAN100への送信や、また、LAN100からのファイルの受信や、更に、それらのファイルの表示および/または処理を行うことができる、一般的なPCにより構成さ

れる。図1にパーソナルコンピュータ機器が示されているが、ネットワークソフトウェアを実行するのに適切であるような、他のコンピュータ機器を含んでもよい。例えば、UNIXのソフトウェアを使用している場合に、UNIXワークステーションをネットワークに含んでも良く、これらのワークステーションは、適切な状況下で、図示されているPCと共に使用することができる。

【0010】通常、LAN100などのLANは、1つの建物内の1つの階または連続した複数の階でのユーザグループ等の、比較的ローカルなユーザグループにサービスを提供する。例えば、ユーザが他の建物や他県に居るなど、あるユーザが他のユーザから離れるに従って、ワイドエリアネットワーク(WAN)を作ってもよい。WANは基本的に、いくつかのLANをサービス総合デジタルネットワーク(ISDN)などの高速度デジタル線で接続することにより形成された集合体である。従って、図1に示すように、LAN100、LAN110、LAN120とは、モデム/トランスポンダー130およびバックボーン140を介して接続されWANを形成する。

【0011】それぞれのLANは、専用のPCを含み、また、必要に応じてファイルサーバやプリントサーバを含むこともある。図1に示すように、LAN110は、PC111、PC112、ファイルサーバ113、ネットワークディスク114、プリントサーバ115、プリンタ116を含む。対照的に、LAN120は、PC121とPC122のみを含む。LAN100、LAN110、およびLAN120に接続されている機器は、WAN接続を介して他のLANに接続された機器の機能にアクセスすることができる。

【0012】このような大規模ネットワークシステムを構成するネットワーク上のデバイスを管理するための方法として、これまでにいくつかの試みが数多くの標準機関でなされている。国際標準化機構(ISO)は開放型システム間相互接続(Open System Interconnection, OSI)モデルと呼ばれる汎用基準フレームワークを提供した。ネットワーク管理プロトコルのOSIモデルは、共通管理情報プロトコル(Common Management Information Protocol, CMIP)と呼ばれる。CMIPはヨーロッパの共通ネットワーク管理プロトコルである。

【0013】また近年では、より共通性の高いネットワーク管理プロトコルとして、簡易ネットワーク管理プロトコル(Simple Network Management Protocol, SNMP)と呼ばれるCMIPに関連する一変種のプロトコルがある。(「TCP/IPネットワーク管理入門実用的な管理をめざして」M.T.ローズ=著/西田竹志=訳(株)トッパン発行1992年8月20日初版を参考)。

【0014】このSNMPネットワーク管理技術によれば、ネットワーク管理システムには、少なくとも1つの

ネットワーク管理ステーション（NMS）、各々がエージェントを含むいくつかの管理対象ノード、及び管理ステーションやエージェントが管理情報を交換するために使用するネットワーク管理プロトコルが含まれる。ユーザは、NMS上でネットワーク管理ソフトウェアを用いて管理対象ノード上のエージェントソフトウェアと通信することにより、ネットワーク上のデータを得、またデータを変更することができる。

【0015】ここでエージェントとは、各々のターゲット装置についてのバックラウンドプロセスとして走るソフトウェアである。ユーザがネットワーク上の装置に対して管理データを要求すると、管理ソフトウェアはオブジェクト識別情報を管理パケットまたはフレームに入れてターゲットエージェントへ送り出す。エージェントは、そのオブジェクト識別情報を解釈して、そのオブジェクト識別情報に対応するデータを取り出し、そのデータをパケットに入れてユーザに送り返す。時には、データを取り出すために対応するプロセスが呼び出される場合もある。

【0016】またエージェントは、自分の状態に関するデータをデータベースの形式で保持している。このデータベースのことを、MIB (Management Information Base)と呼ぶ。MIBは木構造のデータ構造をしており、全てのノードが一意に番号付けされている。このノードの識別子のことを、オブジェクト識別子(OBJECT IDENTIFIER)と呼ぶ。

【0017】このMIBの構造は、管理情報構造 (SMI : Structure of Management Information) と呼ばれ、RFC1155 Structure and Identification of Management Information for TCP/IP-based Internetsで規定されている。

【0018】次に、SNMPプロトコルについて簡単に説明する。ネットワーク管理ユーティリティソフトウェアが動作しているPC(以下、マネージャと呼称する)とSNMPエージェントが動作している管理対象ネットワークデバイス(以下、エージェントと呼称する)とは、SNMPプロトコルを用いて通信を行う。SNMPプロトコルには5種類のコマンドがあり、それぞれGet-request, Get-next-request, Get-response, Set-request, Trapと呼ばれる。

【0019】Get-requestおよびGet-next-requestは、マネージャがエージェントのMIBオブジェクトの値を取得するために、マネージャがエージェントに対して送出するコマンドである。このコマンドを受け取ったエージェントは、MIBの値をマネージャに通知するために、マネージャに対してGet-responseコマンドを送出する。

【0020】Set-requestは、マネージャがエージェントのMIBオブジェクトの値を設定するために、マネージャがエージェントに対して送出するコマンドである。

このコマンドを受け取ったエージェントは、設定結果をマネージャに通知するために、マネージャに対してGet-responseコマンドを送出する。

【0021】Trapは、エージェントが自分自身の状態の変化をマネージャに対して通知するために、エージェントがマネージャに対して送出するコマンドである。

【0022】SNMPエージェントはPCやプリンタ102に接続されているネットワークボード(NB)101上で動作し、SNMPマネージャとなるネットワー

10ク管理ソフトはPCで動作するシステムがよく知られている。しかし近年にのインターネットの普及により、専用のネットワーク管理ソフトをクライアントPC毎に動作させるのではなく、ネットワーク管理ソフトはサーバ上で動作し、さらにユーザインタフェースとしてWWWを用いたシステムもでている。

【0023】PC150ではWWWサーバプログラムが動作しており、PC150のディスクにはHTMLを用いて記述されている多数のWWWページデータが格納されている。PC103上で動作するWWWブラウザプロ

20グラムは、ユーザにより指定されたページを表示するために、PC150上で動作するWWWサーバプログラムに対して指定されたページの取得を要求する。WWWサーバプログラムは、WWWブラウザプログラムからの要求への応答として、指定されたページデータを返す。WWWブラウザプログラムは取得したページデータを解析してその記述に従いページを表示する。

【0024】WWWサーバプログラムは、WWWブラウザプログラムからのページ取得要求中にCGI(Common Gateway Interface)を経由した要求が含まれる場合、CGI

30所定の方法により外部スクリプトやプログラムを起動し、WWWブラウザプログラムからの要求への応答のためのページデータを受け取り、それをWWWブラウザプログラムへ返す。

【0025】次に、上記CGIで起動される外部プログラムが本願のようなネットワーク管理プログラムの場合を説明する。

【0026】WWWサーバプログラムによりCGIで起動されたネットワーク管理プログラムは、SNMPを用いてネットワークに接続されているデバイス、例えばプリンタ102から、管理データを取得する。ネットワーク管理プログラムは取得した管理データをもとにHTMLで記述されたページを生成し、WWWサーバプログラムに返す。

【0027】

【発明が解決しようとする課題】しかし、上記従来のWebブラウザからCGIを経由して外部アプリケーションを起動して、ネットワークワークデバイスの情報を取得するシステムにおいては、アプリケーションがSNMPによる情報取得に失敗した場合、表示する情報がなく

50なってしまう問題があった。

【0028】

【課題を解決するための手段】本発明は上記従来例を鑑みてなされたもので、SNMPによる情報取得に失敗した場合でも何らかの情報を表示することを目的とする。この目的を達成すべく、本発明にかかるネットワークデバイスの情報管理方法及び装置、その方法をコンピュータに実行させるためのプログラムを記録した記憶媒体は主として以下のような構成からなることを特徴とする。

【0029】すなわち、ネットワーク上から所定のインターフェースを経由してサーバのアプリケーションを起動し、取得するネットワークデバイスの情報を管理する方法は、ブラウザがサーバに対して表示すべき情報の取得要求を出すステップと、前記表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行うステップと、前記情報取得の成否を判定するステップと、前記判定に基づき、前記情報取得に成功した場合に、取得した値を表示するステップと、前記取得した値をキャッシュ値として更新するステップと、前記判定に基づき、前記情報取得に失敗した場合に、キャッシュ値として保存されていた既取得情報を表示するステップとを備える。

【0030】また、ネットワーク上から所定のインターフェースを経由してサーバのアプリケーションを起動し、取得するネットワークデバイスの情報を管理する装置は、ブラウザがサーバに対して表示すべき情報の取得要求を出す手段と、前記表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行う手段と、前記情報取得の成否を判定する手段と、前記判定に基づき、前記情報取得に成功した場合に、取得した値を表示する手段と、前記取得した値をキャッシュ値として更新する手段と、前記判定に基づき、前記情報取得に失敗した場合に、キャッシュ値として保存されていた既取得情報を表示する手段とを備える。

【0031】また、ネットワーク上から所定のインターフェースを経由してサーバのアプリケーションを起動し、ネットワークデバイスの情報を取得するプログラムを記憶するコンピュータ可読の記憶媒体であって、該プログラムが、ブラウザがサーバに対して表示すべき情報の取得要求を出すステップのコードと、前記表示すべき情報についてSNMPによる情報取得をネットワークデバイスに対して行うステップのコードと、前記情報取得の成否を判定するステップのコードと、前記判定に基づき、前記情報取得に成功した場合に、取得した値を表示するステップのコードと、前記取得した値をキャッシュ値として更新するステップのコードと、前記判定に基づき、前記情報取得に失敗した場合に、キャッシュ値として保存されていた既取得情報を表示するステップのコードとを備える。

【0032】

【発明の実施の形態】<LANの構成>ネットワーク管

理ソフトウェアが管理するデバイスが接続されているLANについて、以下で説明する。図1は、プリンタをネットワークに接続するためのネットワークボード(NB101)を、開放型アーキテクチャをもつプリンタ102につなげた場合を示す図である。NB101は、ローカルエリアネットワーク(LAN)100に、同軸コネクタをもつEthernetインターフェース10Base-Tや、RJ-45をもつ10Base-TなどのLANインターフェースを介して接続される。

10 【0033】PC103、PC104などの、複数のパーソナルコンピュータ(PC)もまた、LAN100に接続されており、ネットワークオペレーティングシステムの制御のもと、これらのPCはNB101と通信することができる。従って、PCの1つ、例えばPC103を、ネットワーク管理用に使用することができる。また、PCに、PC104に接続されているプリンタ105のようなローカルプリンタを接続してもよい。

【0034】LAN100には、ファイルサーバ106が接続されており、ファイルサーバ106は大容量(例えば、100億バイト)のネットワークディスク107に記憶されたファイルへのアクセスを管理する。

20 【0035】プリントサーバ108は、接続された複数のプリンタ109、または遠隔地にあるプリンタ105などのプリンタに印刷を行わせる。また、他の図示しない周辺機器をLAN100に接続してもよい。

【0036】また、LAN100には、WWWサーバ150が接続されており、WWWサーバ150上にインストールされたネットワーク管理ソフトウェアが生成したHTMLドキュメントを、PC103上にインストールされたWWWブラウザを使って表示したり、PC103上のWWWブラウザ上で行ったプリンタの設定を、WWWサーバ150上のネットワーク管理ソフトウェアを介して特定のプリンタに送信することもできる。

【0037】更に詳しくは、図1に示すネットワークは、様々なネットワークメンバ間で効率良く通信を行うために、NovellやUNIXのソフトウェアなどのネットワークソフトウェアを使用することができる。どのネットワークソフトウェアを使用することも可能であるが、例えば、Novell社のNetWare(Novell40)の登録商標。以下、省略)ソフトウェアを使用することができる。このソフトウェアパッケージに関する詳細な説明は、NetWareパッケージに同梱されているオンラインドキュメンテーションを参照のこと。これは、Novell社からNetWareパッケージと共に購入することができる。

【0038】簡潔に説明すると、ファイルサーバ106は、LANメンバ間でファイルの受信や、記憶、キューイング、キャッシング、および送信を行う、ファイル管理部としての役割を果たす。例えば、PC103およびPC104のそれぞれにより作成されたデータファイル

は、ファイルサーバ106へ送られ、ファイルサーバ106は、これらのデータファイルを順に並べ、そしてプリントサーバ108からのコマンドに従って、並べられたデータファイルをプリンタ109の1つに送信する。

【0039】PC103とPC104は、それぞれデータファイルの生成や、生成したデータファイルのLAN100への送信や、また、LAN100からのファイルの受信や、更に、それらのファイルの表示および／または処理を行うことができる、一般的なPCにより構成される。図1にパーソナルコンピュータ機器が示されているが、ネットワークソフトウェアを実行するのに適切であるような、他のコンピュータ機器を含んでもよい。例えば、UNIXのソフトウェアを使用している場合に、UNIXワークステーションをネットワークに含んでも良く、これらのワークステーションは、適切な状況下で、図示されているPCと共に使用することができる。

【0040】通常、LAN100などのLANは、1つの建物内の1つの階または連続した複数の階でのユーザグループ等の、比較的ローカルなユーザグループにサービスを提供する。例えば、ユーザが他の建物や他県に居るなど、あるユーザが他のユーザから離れるに従って、ワイドエリアネットワーク(WAN)を作ってもよい。WANは、基本的には、いくつかのLANをサービス組合ディジタルネットワーク(ISDN)などの高速度ディジタル線で接続することにより形成された集合体である。従って、図1に示すように、LAN100、LAN110、LAN120とは、モデム／トランスポンダ130およびバックボーン140を介して接続され、WANを形成する。

【0041】それぞれのLANは、専用のPCを含み、また、必要に応じて、ファイルサーバやプリントサーバを含むこともある。図1に示すように、LAN110は、PC111、PC112、ファイルサーバ113、ネットワークディスク114、プリントサーバ115、プリンタ116を含む。対照的に、LAN120は、PC121とPC122のみを含む。LAN100、LAN110、およびLAN120に接続されている機器は、WAN接続を介して他のLANに接続された機器の機能にアクセスすることができる。

【0042】<PCの構成>図2は、ネットワーク管理ソフトウェアが稼動可能なPCの構成を示すブロック図である。図2において、150は、ネットワーク管理ソフトウェアが稼動するPCであり、図1におけるWWWサーバ150と同等である。PC150は、ROM302もしくはハードディスク(HD)311に記憶された、あるいはフロッピー(登録商標)ディスク(FD)312より供給されるネットワーク管理プログラムを実行するCPU301を備え、システムバス304に接続される各デバイスを総括的に制御する。

【0043】303はRAMで、CPU301の主メモ

リ、ワークエリアなどとして機能する。305はキーボードコントローラ(KBC)で、キーボード(KB)309や不図示のポインティングデバイスなどからの指示入力を制御する。306はCRTコントローラ(CRTC)で、CRTディスプレイ(CRT)310の表示を制御する。307はディスクコントローラ(DKC)で、ブートプログラム、種々のアプリケーション、編集ファイル、ユーザファイル、およびネットワーク管理プログラムなどを記憶するハードディスク(HD)311およびフロッピーディスク(FD)312とのアクセスを制御する。308は、ネットワークインターフェースカード(NIC)で、LAN100を介して、エージェントあるいはネットワーク機器と双方向にデータをやりとりする。

【0044】<ネットワーク管理ソフトウェアのモジュール構成>図3は、本発明に係るネットワーク管理ソフトウェアのモジュール構成図である。本発明に係るネットワーク管理ソフトウェア1062は、図2におけるハードディスク311に格納されており、CPU301によって実行される。その際、CPU301はワークエリアとしてRAM303を使用する。

【0045】図3において、ネットワーク管理ソフトウェア1062は、WWWサーバプログラム1061から起動され、CGIインターフェース402を介してCGIパラメータ、およびHTMLドキュメントのやり取りを行う。

【0046】403は全体制御モジュールで、後述のパラメータモジュール404に対してCGIパラメータを登録したのち、CGIパラメータの中のコマンドパラメータに応じて、後述のシステムモジュール405、デバイスリストモジュール407、デバイス詳細モジュール409のいずれかに制御を振り分けるためのモジュールである。CGIパラメータに誤りがあった場合は、後述のテンプレートモジュール412を介して、CGIパラメータに誤りがある旨のHTMLドキュメントを生成することもある。

【0047】404はパラメータモジュールで、全体制御モジュール403により登録されたCGIパラメータをテーブル形式で保存・管理するモジュールである。他のモジュールは、必要に応じて、パラメータモジュール402から、所望するパラメータを取得することができる。

【0048】405はシステムモジュールで、ネットワーク管理ソフトウェア1062の動作を規定するシステムパラメータ(例えば、HTMLドキュメントの自動更新間隔など)の表示・設定を制御し、また、関連するHTMLドキュメントを生成するモジュールである。システムモジュール405は、パラメータモジュール404からコマンドパラメータを取得し、コマンドパラメータの内容がシステムパラメータの表示要求であった場合

は、システム設定ファイル406から必要な情報を読み出し、テンプレートモジュール412を介してシステムパラメータ表示用のHTMLドキュメントを生成する。

【0049】また、取得したコマンドパラメータの内容がシステムパラメータの設定要求であった場合は、システム設定ファイル406に対して通知されたシステムパラメータを書き込み、テンプレートモジュール412を介して、設定後に表示すべきHTMLドキュメントを生成する。

【0050】尚、図には示していないが、システム設定ファイル406に保存したシステムパラメータは、ネットワーク管理ソフトウェア1062を構成する各モジュールが、必要に応じて読み出すことができる。

【0051】407はデバイスリストモジュールで、後述のデバイス探索モジュール408により探索されたデバイスの一覧（デバイスリスト）を示すHTMLドキュメントを生成するためのモジュールである。デバイスリストの表示オプションの処理なども、このデバイスリストモジュール407が制御する。

【0052】408は、デバイス探索モジュールで、ネットワークに接続されたデバイスを探索するモジュールである。

【0053】410はデバイス詳細モジュールで、CGIパラメータにより指定された特定のデバイスに対するより詳細な情報を表示・設定するための制御、および関連するHTMLドキュメントを生成するためのモジュールである。デバイス詳細モジュール410は、指定されたデバイスの詳細な情報を取得・設定するために、指定されたデバイスに対応する後述のデバイス固有モジュール411を使用する。411はデバイス固有モジュールで、ネットワーク管理ソフトウェアの管理対象となるデバイス（プリンタ、ネットワークインターフェースボードなど）ごとに存在する。デバイス固有モジュール411は、表示時においては、デバイスから必要な情報を取得し、取得した情報をテンプレートモジュール412に設定するように動作する。また、設定時においては、CGIパラメータで通知された設定値を、デバイスが解釈可能な値に変換し、デバイスに送信するように動作する。

【0054】409はプロトコルモジュールで、MIB（Management Information Base）のハンドリング、SNMP（Simple Network Management Protocol）パケットの送受信、トранSPORTプロトコルの制御など、ネットワーク管理ソフトウェアがデバイスと通信するために必要な、各種プロトコルの制御を行うためのモジュールである。

【0055】412はテンプレートモジュールで、図2のハードディスク311に保存されているテンプレートファイル413を基に、ネットワーク管理ソフトウェアの出力結果としてのHTMLドキュメントを生成するためのモジュールである。

【0056】テンプレートモジュール412は、CGIパラメータ、全体制御モジュール403、システムモジュール405、デバイスリストモジュール407、またはデバイス詳細モジュール410により指定されたテンプレートファイルをオープンし、テンプレートファイルの内容を解析し、必要に応じてテンプレートファイルに含まれるテンプレート変数を、全体制御モジュール403、システムモジュール405、デバイスリストモジュール407、デバイス詳細モジュール410、またはデ

10 バイス固有モジュール411により設定された値に置き換えることにより、HTMLドキュメントを生成し、CGIインターフェース402を介してWWWサーバプログラムに送信する。HTMLドキュメントを生成する際に使用したテンプレート変数の値、あるいは、生成したHTMLドキュメントファイルは、同一のテンプレートファイルを元にして二回目以降にHTMLドキュメント生成する際の処理時間を短縮するために、キャッシュファイル414として図2のハードディスク311上に保存することもできる。このアプリケーションをWebNetSpotと呼ぶ。

【0057】<<画面の構成>>
<デバイスリスト>ユーザがデバイスリスト画面でURLを指定することによりCGI経由でWebNetSpotが起動される。WebNetSpotは、ネットワークに接続されているデバイスを探査し、探索で検知されたデバイスについて、

- ・デバイスの種類
- ・デバイス名
- ・製品名
- 30 ・ネットワークボード製品名
- ・ネットワークアドレス
- ・デバイスの状態

といった情報をブラウザに表示する。

【0058】デバイスの種類とは、普通のプリンタであるか、それともコピー機能もついた複合機であるかということによって分類され、アイコンで表示される。デバイス名は、ユーザが各デバイスにつけた名前である。

【0059】デバイス名のアイコンをクリックすることにより、WebNetSpotが起動され、このデバイスの詳細情報が表示される。詳細は、後述する。デバイスの状態は、発生しているエラーの重要度によってアイコンが変化する。

【0060】<デバイス詳細>デバイスリストにおいて、デバイス名がクリックされると、WebNetSpotにIPアドレスが渡され、この情報をもとにクリックされたデバイスの詳細の情報が取得され、ブラウザに表示される。表示される情報は、

- ・状態
- ・装備情報
- 50 ・デバイス情報

- ・ネットワークボード情報
- ・プロトコル情報

であり、これらの情報を2画面に表示する。

【0061】図5、図6は、表示画面の一例である。図5には、状態、装備情報、デバイス情報が表示されている。図6には、ネットワークボード情報、プロトコル情報が表示されている。これらのデバイス詳細を表示する画面は、左側のメニュー部分と右側の本体部分に画面の構成が別れる。左側のメニューの画面は、図5、図6で共通の表示構成をもっており、メニュー画面を利用するこにより画面の切り替えを行っている。たとえば、図5の705で示される「ネットワークボード」をクリックすると、画面は図6に切り替わる。

【0062】<WebNetSpotのファイル構成>図7は、WebNetSpotのファイル構成である。2重枠のボックス(901,902,903,904,908,909,910)がディレクトリ、単枠のボックス(905,906,907,911,912,913)はファイルを示している。901は、CGIプログラムとしてのWebNetSpotのルートディレクトリである。この下に、

- ・Document
- ・Images
- ・Template

の各ディレクトリが存在し、さらに、実行ファイルであるWebNetSpot.exe及び各種HTMLファイルが存在する。

【0063】902のDocumentディレクトリの下には、デバイスから取得してきた情報を一時的にキャッシュファイルとして保存するためのものである。903のImagesディレクトリには、情報の表示に使用される各種のイメージファイルが保存されている。904はテンプレートファイルが保存されるディレクトリである。このディレクトリの下には、3種類のディレクトリがある。908で示されるsysディレクトリには、デバイスリスト、エラー関係などのデバイスに依存しない情報の表示のためのテンプレートファイルが格納される。909で示されるproductディレクトリは製品固有の情報に関するテンプレートファイルを格納するところであり、製品の種類の数だけ存在する。

タグ	機能
TEMPLATE	テンプレートの記述
HEAD	ヘッダ記述
BODY	本体記述
LINK	関連するテンプレートファイルの記述
VARIABLE	テンプレート変数の宣言
INCLUDE	テンプレートファイルのインクルード
SET	変数の値の設定
EMBED	変数値の埋め込み
ISVALID	変数値が有効かどうかの評価
EVAL	変数値の比較
LOOP	繰り返しの記述

【0064】このディレクトリに格納される情報は、図5で示される状態、装備情報、デバイス情報である。また、910で示されるNICディレクトリにはネットワークボードに固有の情報が格納され、ネットワークボードの修理の数だけ存在するこのディレクトリに格納される情報は、図6で示されるようなネットワーク情報、プロトコル情報である。なお、複数の製品もしくはネットワークボードの間で、同一のテンプレートファイルを興することが可能な場合は、必ずしも製品もしくはネットワークボードの種類の数だけディレクトリが存在する必要はない。

【0065】<テンプレートファイルの構造>図4を使用して、WebNetSpotで使用されるテンプレートファイルの構造について説明する。テンプレートファイルの内容は、<<TEMPLATE>>タグから<</TEMPLATE>>タグの間に記述される。これは、<<HEAD>>タグと<</HEAD>>タグの間に記述されるHEADブロック、<<BODY>>タグと<</BODY>>タグの間に記述されるBODYブロックの2つから構成される。

【0066】HEADブロックには、主にデバイスから取得すべき情報を変数として記述する<<VARIABLE>>タグが記述される。BODYブロックは、出力されるHTML記述及びデバイスから取得した値を埋め込む<<EMBED>>タグ等から構成される。

【0067】図4において、WebNetSpotは、HEADブロックをパース(解析)することによりWNTVAR_DCVP_PRODUCTによって示されるプロダクト名をデバイスから取得すればよいことがわかる。この情報を取得したWebNetSpotは、SNMP/MIBによりデバイスから情報を取得する。その後、BODYブロックをパースすることにより、WNTVAR_DCVP_PRODUCTの変数を持つ<<EMBED>>タグと、デバイスから取得した情報を置き換える。全ての変数において上記のようなパースを行うことにより、出力すべきHTMLファイルを得ることができる。

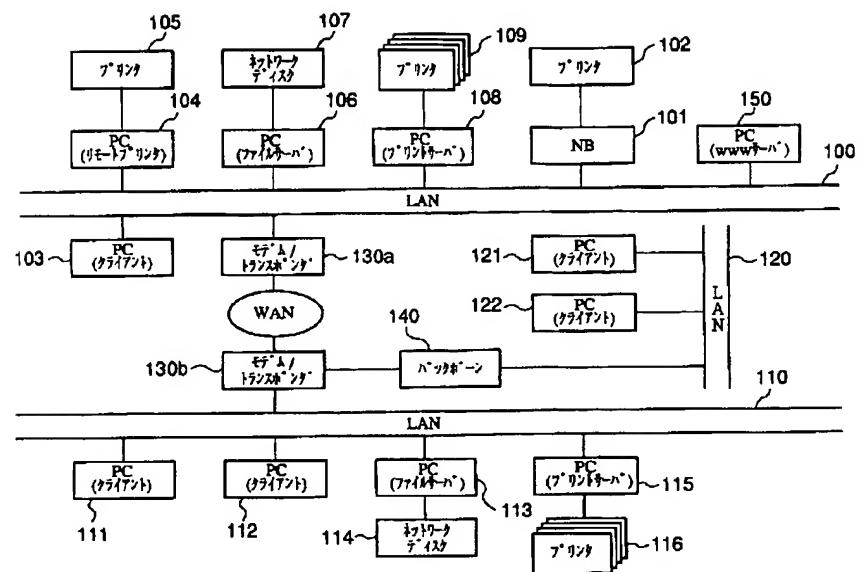
【0068】以下にテンプレートファイルで使用されるタグと機能の一覧を示す。

【0069】

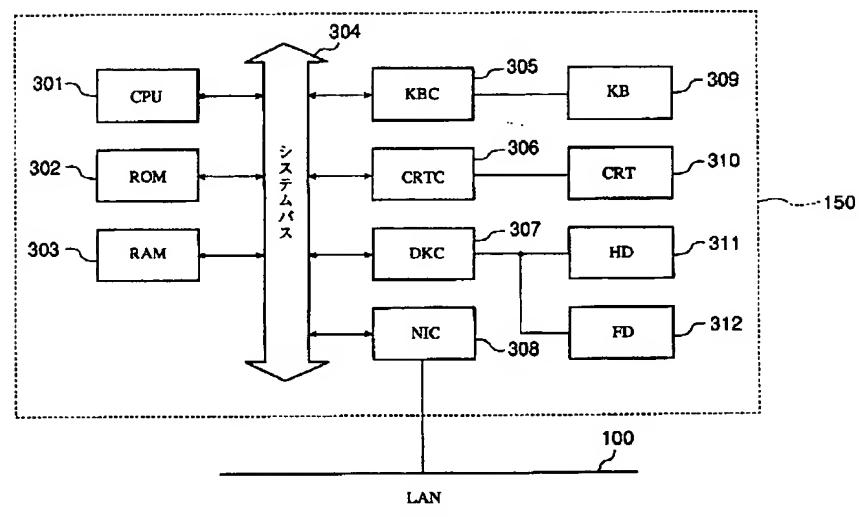
COMMENT	コメントの記述
<キャッシュ値の利用>キャッシュ値は図9で示されるように、変数名、値、最終更新日時をセットとして、個々の情報ごとに保存される。本来キャッシュ値を使用する目的は、SNMPによる情報取得回数を少なくし、上表表示までの時間を短縮することにある。最終更新日時は、情報表示要求が行われた日時と比較され、SNMPによる情報取得を行うかどうかの判断基準として使われる。	が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。
【0070】以下では、キャッシュ値をSNMPによる情報取得の失敗が生じたときにどのように利用されるかの流れを記述する。図8はそのフローチャートである。	【0077】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わる
【0071】ステップS801で、WebブラウザからWebサーバに対して情報表示要求が行われると、アプリケーションが起動され、ステップS802においてSNMPによる情報取得が行われる。この情報取得が成功した場合、ステップS804においてアプリケーションは取得した値をパースしたテンプレートファイルに埋め込み、最終的にHTMLファイルとしてブラウザに送り出す。さらに、ステップS805においてキャッシュファイルにキャッシュ値があればを取得した値に更新しなければそのまま書き込む。	10 メモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。
【0072】一方、情報取得に失敗した場合は、キャッシュ値(S803)をパースしたテンプレートファイルに書き込み表示を行う。この際に、今回取得した値でなく、キャッシュ値を表示していることを示すために、表示する文字列の色や書体を変えることも可能である。SNMPによる情報取得に失敗した場合は、以前に情報の取得に成功し、キャッシュ値として更新された値を表示に用いることにより、ユーザは少なくとも前回取得した値を参照することが可能となる。	【0078】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図3に示す各モジュールに対応したプログラムコードを記憶媒体に格納することになる。
【0073】	20 【0079】 【発明の効果】以上説明したように、本発明によれば、SNMPによる情報取得に失敗した場合は、以前に情報の取得に成功し、キャッシュ値として保存された既取得情報を表示に用いることにより、ユーザは少なくとも前回取得した値を参照することが可能となる。
【他の実施形態】なお、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成されることは言うまでもない。	【0080】 【図面の簡単な説明】 【図1】LANの構成図である。 【図2】ネットワーク管理ソフトウェアが稼動可能なPCの構成図である。
【0074】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。	30 【図3】ネットワーク管理ソフトウェアのモジュール構成図である。 【図4】テンプレートファイルの例を示す図である。 【図5】デバイスリストの例を示す図である。 【図6】デバイスリストの例を示す図である。 【図7】ネットワーク管理ソフトウェアのファイル構成図である。
【0075】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。	40 【図8】本発明の実施形態に係るキャッシュファイルを利用した情報表示方法に関するフローチャートである。 【図9】本発明の実施形態に係るキャッシュファイルの記述例を示す図である。
【0076】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能	50 【符号の説明】 100 ローカルエリアネットワーク（LAN） 101 ネットワークボード（NB） 102 開放型アーキテクチャを持つプリンター 103 LAN100に接続されているパーソナルコンピュータ（PC） 104 LAN100に接続されているパーソナルコンピュータ（PC） 105 PC104に接続されているプリンター

106 LAN100に接続されているファイルサーバー	310 PC200のCRTディスプレイ(CRT)
—	311 PC200のハードディスク(HD)
107 LAN100に含まれるネットワークディスク	312 フロッピーディスクドライブ(FD)
108 LAN100に含まれるプリントサーバー	402 CGIインタフェース
109 プリントサーバー108に接続されたプリンタ	403 全体制御モジュール
—	404 パラメータモジュール
110 ローカルエリアネットワーク(LAN)	405 システムモジュール
111 LAN110に接続されているパーソナルコンピュータ(PC)	406 システム設定ファイル
112 LAN110に接続されているパーソナルコンピュータ(PC)	407 デバイスリストモジュール
113 LAN110に接続されているファイルサーバー	10 408 デバイス探索モジュール
—	409 デバイス詳細モジュール
114 LAN110に含まれるネットワークディスク	410 デバイス固有モジュール
115 LAN110に含まれるプリントサーバー	411 プロトコルモジュール
116 プリントサーバー110に接続されたプリンタ	412 テンプレートモジュール
—	413 テンプレートファイル
120 ローカルエリアネットワーク(LAN)	414 キャッシュファイル
121 LAN120に接続されているパーソナルコンピュータ(PC)	901 ネットワーク管理ソフトウェアWebNetSpotを格納するWebNetSpotディレクトリ
122 LAN120に接続されているパーソナルコンピュータ(PC)	902 キャッシュファイルを格納するDocumentディレクトリ
130a 変調／復調(MODEM)／トランスポンダ	20 903 画像ファイルを格納するImagesディレクトリ
—	904 テンプレートファイルを格納するTemplateディレクトリ
130b 変調／復調(MODEM)／トランスポンダ	905 ネットワーク管理ソフトウェアWebNetSpot
—	906 HTMLファイル
140 バックボーン	907 画像ファイル
150 LAN100に接続されWWWサーバが動作しているパーソナルコンピュータ(PC)	908 共通使用されるテンプレートファイルを格納するSYSディレクトリ
301 FD212より供給されるネットワーク管理プログラムを実行するCPU	909 デバイス依存するテンプレートファイルを格納するPRODUCTディレクトリ
302 PC200のROM	30 910 ネットワークボード依存するテンプレートファイルを格納するNICディレクトリ
303 PC200のRAM	911 テンプレートファイル
304 PC200のシステムバス	912 テンプレートファイル
305 PC200のキーボードコントローラ(KB C)	913 テンプレートファイル
306 PC200のCRTコントローラ(CRTC)	1031 WWWブラウザプログラム
307 PC200のディスクコントローラ(DKC)	1051 WWWサーバプログラム
308 PC200のネットワークインタフェースカード(NIC)	1052 ネットワーク管理プログラム
309 PC200のキーボード(KB)	1061 WWWサーバプログラム
	40 1062 ネットワーク管理ソフトウェア

【図1】



【図2】



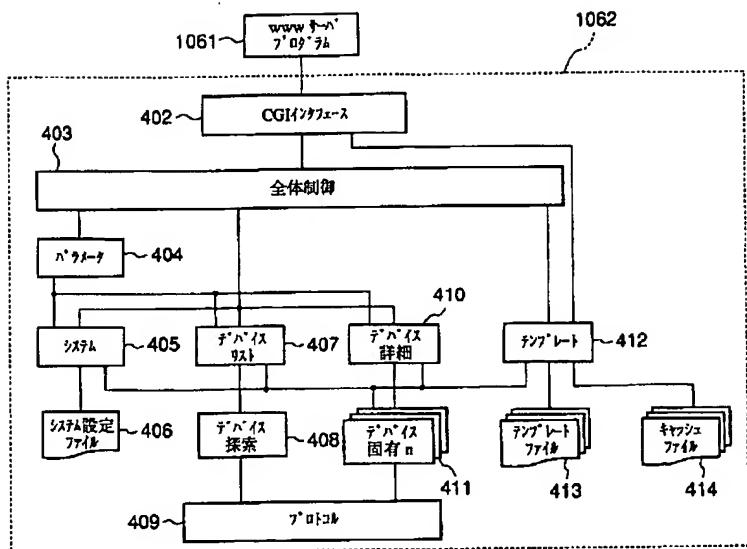
【図9】

キャッシュファイルの例

```

<NAME=" WNTVAR.DCV.PRODUCT" VALUE=" LBP" TIME=" 6, 2, 10, 30 >
<NAME=" WNTVAR.DCV.NETADDR" VALUE=" 192.168.18.30" TIME=" 6, 2, 9, 6 >
<NAME=" WNTVAR.DCV.STATUS" VALUE=" ready" TIME=" 6, 2, 10, 35 >
:
:
:
  
```

【図3】



【図4】

テンプレートファイルの構造

<TEMPLATE>

<HEAD>
HEAD ブロックの記述

</HEAD>

<BODY>
BODY ブロックの記述

</BODY>

</TEMPLATE>

HEAD ブロックの記述

```

<HEAD>
  <VARIABLE NAME="WNTVAR.DCV.PRODUCT" EXP="1.0.0">
  <VARIABLE NAME="WNTVAR.DCV.NETADDR" EXP="0.1.0">
  <VARIABLE NAME="WNTVAR.DCV.STATUS" EXP="0.0.10">
  ...
</HEAD>>

```

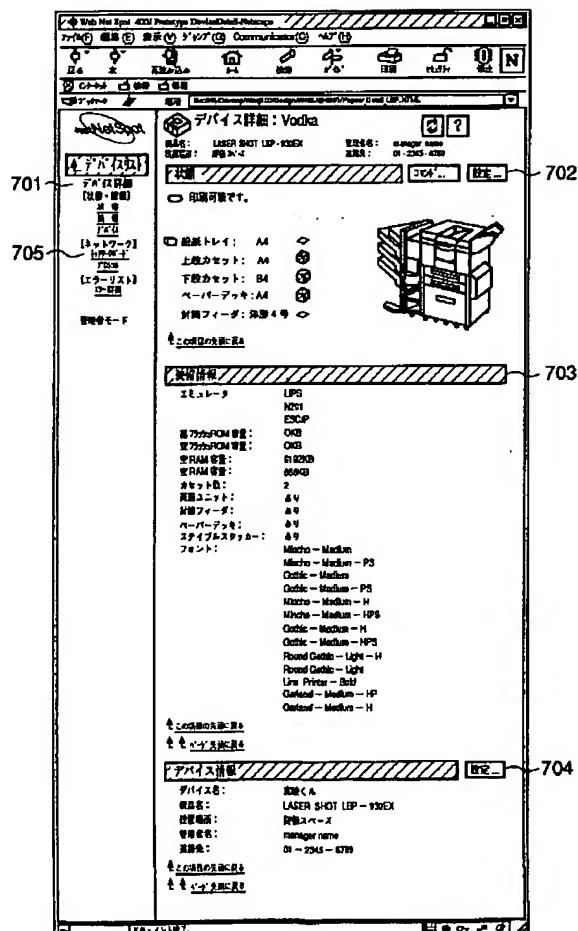
BODY ブロックの記述

```

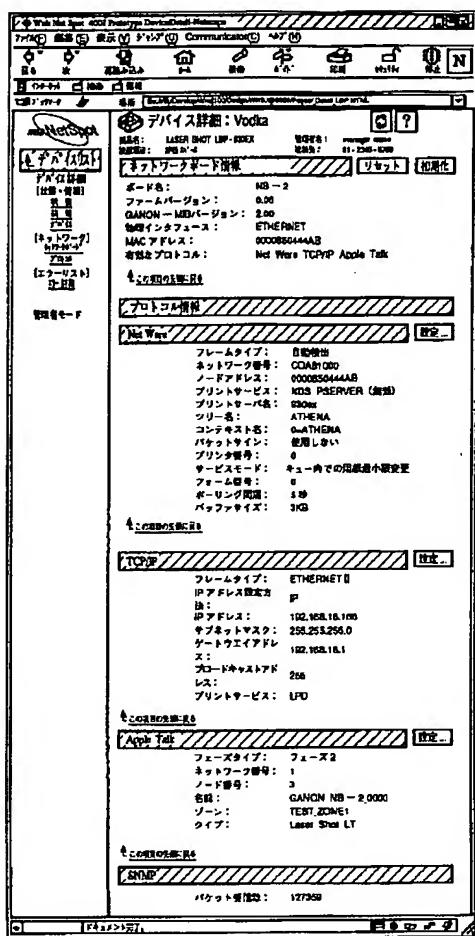
<BODY>
  <HTML>
    <BODY>
      ...
      <TD><FONT SIZE="3">
        <EMBEDDED NAME="WNTVAR.DCV.PRODUCT">
      </FONT></TD>
      ...
    </BODY>
  </HTML>
</BODY>

```

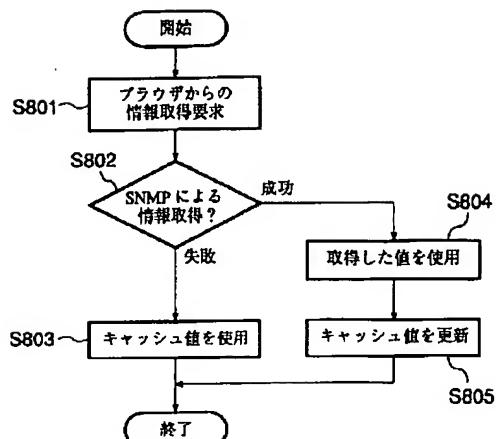
【図5】



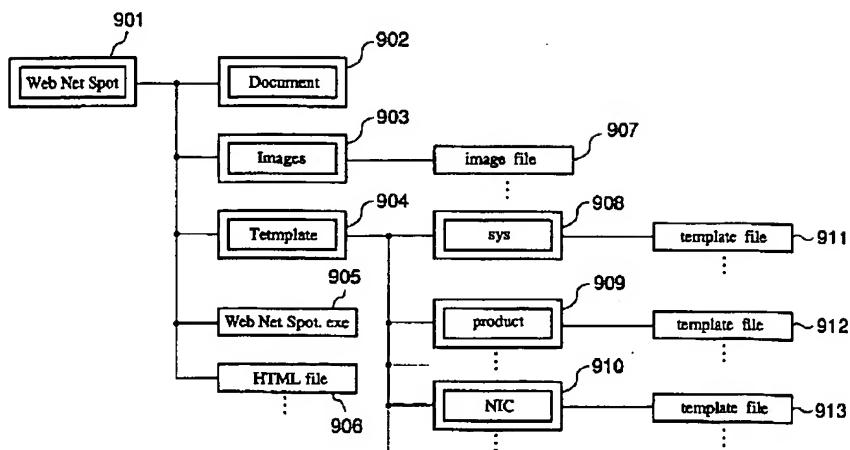
【図 6】



【図 8】



【図 7】



フロントページの続き

F ターム(参考) 5B085 AC03 AC13 AC16 BG07
5B089 GA21 GB04 GB08 GB09 HA06
HB06 JA35 JB23 KA12 KB04
KC44 KD02 KF03 LB23 MD04
ME09
5E501 AA13 AB15 AC25 AC35 BA03
BA20 CA02 DA01 EA34 FA13
FA46 FB26 FB28 FB46

10

(11) Japanese Patent Laid-Open No. 2000-148632

(43) Laid-Open Date: May 30, 2000

(21) Application No. 10-324044

(22) Application Date: November 13, 1998

(71) Applicant: CANON KABUSHIKI KAISHA

(72) Inventor: Tetsuya YAMADA

(74) Agent: Patent Attorney, Yasunori OTSUKA

(54) [Title of the Invention] INFORMATION MANAGING METHOD
OF NETWORK DEVICE, APPARATUS AND STORAGE MEDIUM

(57) [Abstract]

[Problem] In a system for acquiring information about network devices by activating an external application via a CGI from a Web browser, when the application fails to acquire information by SNMP, a problem is encountered in that there is no more information to be displayed.

[Solving Means] The browser issues an acquisition request of information to be displayed to a server (S801); conducts information acquisition regarding the network devices by means of SNMP to get information to be displayed; and determines whether or not the acquisition is successful (S802). When acquisition is successful, the value of acquisition is displayed (S804), and the value is updated as a cache value (S805). If information acquisition is not

successful, an already acquired information stored as a cache value is displayed (S803).

[Claims]

[Claim 1]

A management method of information about network devices acquired by activating a server application via a prescribed interface from a network, comprising:

a step of issuing an acquisition request of information to be displayed by a browser to the server;

a step of performing information acquisition by SNMP to network devices regarding said information to be displayed;

a step of determining whether or not said information acquisition is successful;

a step of displaying the acquired value when said information acquisition is successful according to said determination;

a step of updating said acquired value as a cache value; and

a step of displaying already acquired information stored as a cache value when said information acquisition is not successful according to said determination.

[Claim 2]

A management apparatus of information about network devices acquired by activating a server application via a prescribed interface from a network, comprising:

means of issuing an acquisition request of information to be displayed by a browser to a server;

means of performing information acquisition by SNMP to network devices regarding said information to be displayed;

means of determining whether or not said information acquisition is successful;

means of displaying the acquired value when said information acquisition is successful according to said determination;

means of updating said acquired value as a cache value; and

means of displaying already acquired information stored as a cache value when said information acquisition is not successful according to said determination.

[Claim 3]

A computer-readable storage medium storing a program for acquiring information about network devices by activating a server application via a prescribed interface from a network, said program comprising:

a code of a step of issuing an acquisition request of information to be displayed by a browser to a server;

a code of a step of performing information acquisition by SNMP to network devices regarding information to be displayed;

a code of a step of determining whether or not said information acquisition is successful;

a code of a step of displaying the acquired value when

said information acquisition is successful according to said determination;

 a code of a step of updating said acquired value as a cache value; and

 a code of a step of displaying already acquired information stored as a cache value when said information acquisition is not successful according to said determination.

[Claim 4]

 The information management method of network devices according to claim 1, wherein said network devices include a printer.

[Claim 5]

 The information management method of network devices according to claim 1, wherein said prescribed interface is a common gateway interface.

[Claim 6]

 The information management apparatus of network devices according to claim 2, wherein said network devices include a printer.

[Claim 7]

 The information management apparatus of network devices according to claim 2, wherein said prescribed interface is a common gateway interface.

[Claim 8]

The information management method of network devices according to claim 1, wherein said step of displaying already acquired information stored as a cache value conducts display control for discriminating a case where information acquisition is successful.

[Claim 9]

The information management method of network devices according to claim 8, wherein said display control makes discrimination by changing the characters or color of a character string.

[Claim 10]

The information management apparatus of network devices according to claim 2, wherein said means of displaying the already acquired information stored as a cache value performs display control for discriminating from a case of successful acquisition of information.

[Claim 11]

The information management apparatus of network devices according to claim 10, wherein said display control accomplishes discrimination by changing characters or color of the character string.

[Detailed description of the Invention]

[0001]

[Technical Field]

The present invention relates to a computer network.

More particularly, the invention relates to an information management method managed by network management software, an apparatus for executing such a method, and a computer-readable storage medium storing a program for executing this information management method on a computer.

[0002]

[Background Art]

Fig. 1 illustrates a case where a network board (NB) 101 for connecting a printer to a network is connected to a printer 102 having an open architecture. NB 101 is connected to a local area network (LAN) via a LAN interface such as an Ethernet interface 10Base-2 having a coaxial connector or a 10Base-T having an RJ-45.

[0003] A plurality of personal computers (PC) such as PC 103 and PC 104 are also connected to the LAN 100 and can communicate with NB 101 under control of the network operating system. Therefore, one of these PCs, for example, the PC 103 can be used for network management. A local printer such as a printer 105 connected to the PC 104 may be connected to the PC.

[0004] A file server 106 is connected to the LAN 100. The file server 106 manages access to files stored in a large-capacity (for example, 10 giga-bytes) network disk 107.

[0005] The print server 108 causes a plurality of connected printers 109, or a printer 105 at a remote place to conduct

printing. A peripheral device not shown may be connected to the LAN 100.

[0006] A www server 150 is connected to the LAN 100. HTML documents generated by network management software installed on the www server 150 can be displayed by the use of a www browser installed on the PC 103, or setting of the printer performed on the www browser on the PC 103 can be transmitted to a specific printer via network management software on the www server 150.

[0007] More specifically, the network shown in Fig. 1 can use network software such as Novell or UNIX software to ensure high-efficiency communication between various network members. While it is possible to use any software, it is possible to use, for example, Novell's NetWare (Novell's registered trademark, this note is hereinafter omitted) software. For detailed explanation regarding this software package, refer to Online Documentation enclosed in the NetWare package. This is commercially available from Novell together with the NetWare package.

[0008] Briefly, the file server 106 plays the role of a file management unit that performs file receiving, storage, queuing, caching, and transmission between LAN members. For example, data files prepared individually by the PC 103 and PC 104 are sent to the file server 106. The file server 106 arranges these data files in sequence, and transmits the

arranged data files to one of the printers 109 in accordance with a command from the print server 108.

[0009] The PC 103 and the PC 104 are each composed of a general-purpose PC that can generate data files, respectively, transmit generated data files to the LAN 100, receive files from the LAN 100, and display and/or process these files. Apart from the personal computer equipment shown in Fig. 1, other computer equipment suitable for executing the network software may be included. For example, when UNIX software is used, UNIX workstations may be included in the network. These workstations are applicable with the PCs shown under appropriate circumstances.

[0010] A LAN such as the LAN 100 provides services to a relatively local user group including a user group working on a floor or on a plurality of contiguous floors of a building. For example, according as a user becomes more distant from the other users including cases where the former is in a different building or prefecture, a wide area network (WAN) may be organized. A WAN basically means an aggregate formed by connecting several LANs with high-speed digital lines such as integrated service digital networks (ISDN). As shown in Fig. 1, therefore, a WAN is formed by connecting to the LAN 100, a LAN 110 and a LAN 120 via a modem/transponder 130 and a backbone 140.

[0011] Each LAN includes a dedicated PC, and as required

may include a file server or a print server. As shown in Fig. 1, the LAN 110 includes a PC 111, a PC 112, a file server 113, a network disk 114, a print sever 115, and a printer 116. In contrast, the LAN 120 includes only a PC 121 and a PC 122. Devices connected to the LAN 100, the LAN 110 and the LAN 120 can access the functions of the devices connected to the other LANs via WAN connections.

[0012] Many standardization organizations have made some trials to find methods for managing devices on a network composing such a large-scale network. The International Standardization Organization (ISO) provided a general-purpose standard framework known as the Open system Interconnection (OSI) model. The OSI model of the network management protocol is called the Common Management Information Protocol (CMIP). CMIP is a common network management protocol for Europe.

[0013] More recently, a variation of protocol relating to CMIP called the Simple Network Management Protocol (SNMP) is available as a more common network management protocol.

(Refer to M.T. rose: "Aiming at Practical Management: Introduction to TCP/IP Network Management" translated by T. Nishida, published by Toppan Co., 1st ed. August 20, 1992).

[0014] According to this SNMP network management technology, a network management system includes at least a network management station (NMS), several management target nodes

each including an agent, and a network management protocol used by the management station or the agent to exchange management information. The user can get data on the network and change data by communicating with the agent software on the management target node, using the network management software on the NMS.

[0015] The term "agent" as used herein means software running as a background process for each target device. When the user requests management data to devices on the network, the management software places object identifying information in a management packet or a frame and sends it out to the target agent. The agent interprets the object identifying information, takes out data corresponding to that object identifying information, and sends it back in a packet to the user. Sometimes, a process corresponding thereto may be invoked to take out the data.

[0016] The agent holds data about its status in the form of a database. This database is called an MIB (Management Information Base). An MIB has a tree structure, and all the nodes are unitarily numbered. The identifier of this node is referred to as an object identifier.

[0017] This structure of MIB is called the Structure of Management Information (SMI) and set forth in the "RFC1155 Structure and Identification of Management Information for TCP/IP-based Internets."

[0018] The SNM protocol will now be briefly described. A PC on which the network management utility software is operating (hereinafter referred to as a "manager") and a management target network device on which the SNMP agent is working (hereinafter referred to as an "agent") communicate with each other by use of the SNMP protocol. The SNMP protocol has five kinds of command which are called Get-request, Get-next-request, Get-response, and Set-request.

[0019] Get-request and Get-next-request are commands sent by the manager to an agent for the manager to get a value of MIB object of the agent. Upon receipt of this command, the agent sends out Get-response command to the manager to notify the manager of this value of MIB.

[0020] Set-request is a command sent out by the manager to the agent for the manager to set a value of MIB object of the agent. Upon receipt of this command, the agent sends the Get-request command to the manager to notify the manager of the result of setting.

[0021] Trap is a command for sending by the agent to the manager for the agent to notify the manager of a change in its own status.

[0022] The SNMP agent operates on a network board (NB) 101 connected to a PC or a printer 102, a network management software serving as an SNMP manager operating on a PC is well known. However, under the influence of recent

popularization of the internet, systems are now available, in which network management software runs on a server and www is used as a user interface, not that the special network management software operates for each client PC.

[0023] A www server program operates on the PC 150, and many page data expressed in HTML are stored in the disk of the PC 150. The www browser program operating on the PC 103 requests acquisition of a designated page to the www server program running on the PC 150. In response to the request from the www browser program, the www server program returns the designated page. The www browser program analyzes the acquired page data and displays the page in accordance with the description.

[0024] When a page acquisition request from the www browser program contains a request having passed through a CGI (Common Gateway Interface), the www server program activates an external script or a program by a method prescribed by a CGI, receives page data for responding to the request from the www browser program, and returns it back to the www browser program.

[0025] Then, a case where the external program activated by a CGI is a network management program as in the present application will now be described.

[0026] The network management program activated by the CGI according to the www server program gesticulates the management

data from a device connected to the network by use of the SNMP such as the printer 102. The network management program generates the page described in HTML on the basis of the acquired management data, and returns it back to the www server program.

[0027]

[Problems to be Solved by the Invention]

However, in a system in which information of a network device is acquired by activating an external application via the CGI from the above-mentioned conventional Web browser, if the application fails to get information by SNMP, a problem occurs in that there is no information to be displayed.

[0028]

[Means for Solving the Problems]

The present invention was developed in view of the above-mentioned conventional case and has an object to permit displaying some information even in failure of information acquisition based on SNMP. To achieve this object, the information management method of network devices of the present invention and the apparatus thereof, and the storage medium recording a program for causing a computer to execute this method have substantially the following configuration.

[0029] More specifically, the management method of

information about network devices acquired by activating a server application via a prescribed interface from a network of the present invention comprises a step of issuing an acquisition request of information to be displayed by a browser to the server; a step of performing information acquisition by SNMP to network devices regarding the information to be displayed; a step of determining whether or not the information acquisition is successful; a step of displaying the acquired value when the information acquisition is successful according to the determination; a step of updating the acquired value as a cache value; and a step of displaying already acquired information stored as a cache value when the information acquisition is not successful according to the determination.

[0030] The apparatus for managing information of a network device acquired by activating a server application via a prescribed interface on the network of the present invention comprises means of issuing an acquisition request of information to be displayed by a browser to a server; means of performing information acquisition by SNMP to network devices regarding the information to be displayed; means of determining whether or not the information acquisition is successful; means of displaying the acquired value when the information acquisition is successful according to the determination; means of updating the acquired value as a

cache value; and means of displaying already acquired information stored as a cache value when the information acquisition is not successful according to the determination.

[0031] The present invention provides also a computer-readable storage medium storing a program for acquiring information about network devices by activating a server application via a prescribed interface from a network, this program comprising a code of a step of issuing an acquisition request of information to be displayed by a browser to a server; a code of a step of performing information acquisition by SNMP to network devices regarding information to be displayed; a code of a step of determining whether or not the information acquisition is successful; a code of a step of displaying the acquired value when the information acquisition is successful according to the determination; a code of a step of updating the acquired value as a cache value; and a case of a step of displaying already acquired information stored as a cache value when the information acquisition is not successful according to the determination.

[0032]

[Embodiments] <LAN configuration> The LAN to which the device to be managed by the network management software is connected will now be described. Fig. 1 illustrates a case where a network board (NB) 101 for connecting a printer to a

network is connected to a printer 102 having an open architecture. The NB 101 is connected to a local areas network (LAN) 100 via a LAN interface such as an Ethernet interface 10Base-2 having a coaxial connector or a 10Base-T having an RJ-45.

[0033] The plurality of personal computers (PC) such as the PC 103 and the PC 104 are as well connected to the LAN 100, and these PCs can communicate with the NB 101 under control of the network operating system. One of the PCs, for example, the PC 103 can therefore be used for network management. A local printer such as the printer 105 connected to the PC 104 may be connected to a PC.

[0034] A file server 106 is connected to the LAN 100. The file server 106 manages access to files stored in the large-capacity (for example 10 giga-bytes) network disk 107.

[0035] The print server 108 causes the plurality of connected printers 109 or a printer 105 in a remote place to conduct printing. Other peripheral devices not shown may also be connected to the LAN 100.

[0036] The WWW server 150 is connected to the LAN 100. It is therefore possible to display HTML documents generated by the network software installed on the WWW server 150 by means of a WWW browser installed on the PC 103, or transmit a setting of a printer performed on the WWW browser on the PC 103 to a specific printer via the network management

software on the WWW server 150.

[0037] In further detail, the network shown in Fig. 1 can use network software such as Novell or UNIX software for the purpose of efficiently carrying out communication between various network members. While any network software is applicable, Novell's NetWare (a registered trademark of Novel Co., this note will hereinafter be omitted) software can be used. For a detailed explanation of this software package, refer to the online documentation enclosed in the NetWare package. This is commercially available from Novell Co. together with the NetWare package.

[0038] Briefly, the file server 106 performs receiving of files between LAN members, storage, queuing, caching and transmission, thus playing the role of a file management unit. For example, data files prepared respectively by the PC 103 and the PC 104 are sent to the file server 106. The file server 106 arranges these data files in sequence, and transmits the thus arranged data files to one of the printers 109 in accordance with a command from the print server 108.

[0039] The PC 103 and the PC 104 comprise common PCs which can generate data files, transmits generated data files to the LAN 100, receive files from the LAN 100, and perform display and/or processing of these files. Fig. 1 illustrates personal computer devices. Other computer

devices which are appropriate for executing the network software may be included. For example, when using a UNIX software, a UNIX workstations may be included in the network. The workstations are applicable together with the PC shown under appropriate circumstances.

[0040] A LAN such as the LAN 100 usually provides services to relatively local user group such as one occupying one floor or a plurality of contiguous floors in a building. For example, according as the user becomes more distant from the other users by being in another building or in another prefecture, a wide area network (WAN) may be built. A WAN is basically an aggregate formed by connecting several LANs via a high-speed line such as an integrated service digital network (ISDN). Therefore, as shown in Fig. 1, it is connected to the LAN 100, the LAN 110 and the LAN 120 via a modem/transponder 130 and a backbone 140, thus forming a WAN.

[0041] Each LAN has its own PC, and as required may include a file server or a print server. As shown in Fig. 1, the LAN 110 includes the PC 111, the PC 112, the file server 113, the network disk 114, the print server 115, and a printer 116. In contrast, the LAN 120 comprises only the PC 121 and the PC 122. The devices connected to the LAN 110 and the LAN 120 can access functions of devices connected to the other LANs via the WAN connections.

[0042] <PC configuration> Fig. 2 is a block diagram

illustrating the PC configuration on which the network management software is operable. In Fig. 2, reference numeral 150 represents a PC on which the network management software operates, and is equivalent to WWW server 150 in Fig. 1. The PC 150 has a CPU which executes the network management program stored in a ROM 302 or a hard disk (HD) 311, or supplied by a Floppy (registered trademark) disk (FD) 312, and comprehensively controls all the devices connected to a system bus 304.

[0043] Reference numeral 303 represents a RAM which serves as a main memory or a work area for the CPU 301; 305 represents a keyboard controller (KBC) which controls input instructions from the keyboard (KB) 309 or a pointing device not shown; 306 represents a CRT controller (CRTC) which controls display of the CRT display (CRT) 310; 307 represents a disk controller (DKC) which controls access to the hard disk (HD) 311 storing a boot program, various applications, editing files, user files, and the network management program, and a floppy disk (FD) 312; and 308 represents a network interface card (NIC) which performs two-way exchange of data with the agent or a network device via the LAN 100.

[0044] <Module configuration of network management software> Fig. 3 is a module configuration view of the network management software of the present invention. The

network management software 1062 of the present invention is stored in the hard disk 311 shown in Fig. 2 and executed by the CPU 301. The CPU 301 uses a RAM 303 as a work area.

[0045] In Fig. 3, the network management software 1062 is activated by a WWW server program 1061 and performs exchange of CGI parameters and HTML documents via a CGI interface 402.

[0046] Reference numeral 403 represents an overall control module which allocates control to any of a system module 405, a device list module 407, and a device details module 409 described later in response to the command parameter in the CGI parameters, after registering CGI parameters in a parameter module 404 described later. When CGI parameters contain any error, an HTML document to the effect that there is an error in the CGI parameters may be generated via a template module 412 described later.

[0047] Reference numeral 404 represents the parameter module which stores and manages CGI parameters registered by the overall control module 403 in a table form. The other modules can get desired parameters from the parameter module 402 as required.

[0048] Reference numeral 405 represents a system module which controls display and setting of system parameters regulating operation of the network management software 1062, and generates related HTML documents. The system module 405 gets command parameters from the parameter module 404, and

when a command parameter is a display request of system parameters, reads out necessary information from a system setting file 406, and generates an HTML document for system parameter display via the template module 412.

[0049] If the acquired parameter is a setting request of system parameters, the system module writes in notified system parameters in the system setting file 406, and generates an HTML document to be displayed after setting via the template module 412.

[0050] Although not shown, the system parameters stored in the system setting file 406 can be read out as required by any module composing the network management software 1062.

[0051] Reference numeral 407 represents a device list module which generates an HTML document showing a list of devices searched for by a device search module 408 (device list) described later. Processing of the display option of the device list is as well controlled by the device list module 407.

[0052] Reference numeral 408 represents a device search module which searches for a device connected to the network.

[0053] Reference numeral 410 represents a device details module which performs control for displaying and setting more detailed information about a specific device designated by a CGI parameter, and generates related HTML documents. The device details module 410 uses a device intrinsic module

411 described later corresponding to a designated device to get and set detailed information of the designated device; 411 represents the device intrinsic module which is provided for each management target device (the printer, the network interface board, or the like) of the network management software. The device intrinsic module 411 operates, during display, to get necessary information from devices and to set the acquired information in the template module 412. At the time of setting, the device intrinsic module 411 operates so as to convert the setting value notified with CGI parameters into values interpretable by the device and transmit it to the device.

[0054] Reference numeral 409 represents a protocol module which controls various protocols necessary for permitting communication of the network management software with devices including handling of MIB (Management Information base), transmission and receiving of SNMP (Simple Network Management Protocol) packets, and control of the transport protocol.

[0055] Reference numeral 412 represents a template module which generates an HTML document as a result of output of the network management software on the basis of the template file 413 stored in the hard disk 311 shown in Fig. 2.

[0056] The template module 412 opens a template file designated by the CGI parameters, the overall control module

403, the system module 405, the device list module 407, or the device details module 410, analyzes the contents of the template file, and as required, replaces a template variable contained in the template file into a value set by the overall control module 403, the system module 405, the device list module 407, the device details module 410, or the device intrinsic module 411, thereby generating an HTML document, and transmitting it to the WWW server program via the CGI interface 402. The value of the template variable used when generating the HTML document, or the generated HTML document file can be stored on the hard disk 311 shown in Fig. 2 as a cache file 414 to reduce the processing time upon generating the second and subsequent HTML documents on the basis of the same template file. This application is referred to as the WebNetSpot.

[0057] <>Screen configuration>>

<Device list> The WebNetSpot is activated via CGI by the user specifying URL on the device list screen. The WebNetSpot searches for devices connected to the network, and for the device detected through searching, displays the following information on the browser:

- Kind of device,
- Device name,
- Product name,
- Network board product name,

- Network address, and
- State of device.

[0058] The kind of device is based on a classification depending upon whether the device is a usual printer or a composite machine having the copying function, and displayed with an icon. The device name is a name that the user gives to each device.

[0059] By clicking the icon of the device name, the WebNetSpot is activated, and detailed information of the device is displayed. Details will be described later. The state of the device is shown by a change in icon in response to the information of an occurring error.

[0060] <Details of device> When clicking the device name in the device list, the IP address is passed to the WebNetSpot, thus enabling the user to get detailed information about the clicked device on the basis of this information. The detailed information is displayed on the browser. The displayed information includes:

- State,
- Setup information,
- Device information,
- Network board information, and
- Protocol information,

And these items of information are displayed in two screens.

[0061] Figs. 5 and 6 illustrate typical examples of display

screen. Fig. 5 displays the state, the setup information and the device information. Fig. 6 displays the network board information and the protocol information. The screen displaying these device details is divided into a menu portion to the left and a main body portion to the right. The left menu screen portion has a common display configuration in Figs. 5 and 6. The screen portions are switched over by using the menu screen. For example, by clicking the "Network Board" represented by 705 in Fig. 5, the screen is switched over to Fig. 6.

[0062] <File configuration of WebNetSpot> Fig. 7 shows the file configuration of WebNetSport. Double-framed boxes (901, 902, 903, 904, 908, 909 and 910) represent directories and single-framed boxes (905, 906, 907, 911, 912 and 913) represent files. Reference numeral 901 represents a root directory of WebNetSpot as a CGI program. There are the following directories thereunder:

- Document,
- Images, and
- Template,

And further, there are the WebNetSpot.exe which is an execution file and various HTML files.

[0063] The document directory 902 stores information acquired from devices temporarily as cache files; reference numeral 903 represents an images directory which stores

various image files used for information display; 904 represents a directory storing template files. There are three kinds of directory under this directory; the sys directory represented by 908 stores template files for displaying information not dependent upon the device such as those relating to the device list and error; the product directory represented by 909 stores template files regarding information intrinsic to products. These directories are existent in a number equal to the number of kinds of product.

[0064] This directory stores information about the state, setup information and device information shown in Fig. 5. The NIC directory represented by 910 stores information intrinsic to the network board. Information stored in this directory provided in a number equal to the number of repairs of the network board include network information and protocol information as shown in Fig. 6. If a plurality of products or network boards can share a single template file, directories are not necessarily required to be existent in a number equal to that of products or network boards.

[0065] <Structure of template file> The structure of the template file used for WebNetSpot by the use of Fig. 4 will now be described. The contents of a template file are described between the <<TEMPLATE>> and the <</TEMPLATE>> tag. The description comprises a HEAD block described between the <<HEAD>> tag and the <</HEAD>> tag and a BODY block

\

described between the <<BODY>> and the <</BODY>> tag.

[0066] The <<VARIABLE>> tag describing information to be acquired mainly from the device as a variable is described in the HEAD block. The BODY block comprises output HTML description and <<EMBED>> tags for embedding values acquired from devices.

[0067] In Fig. 4, it is suggested that it suffices for WebNetSpot to get from the device the product name shown by WNTVAR_DCV_PRODUCT by parsing (analyzing) the HEAD block. The WebNetSpot having acquired the information gets information from the device by means of SNMP/MIB. Then, the <<EMBED>> tag having a variable of WNTVAR_DCV_PRODUCT is replaced with the information acquired from the device by parsing the BODY block. The HTML file to be outputted can be obtained by performing the above-mentioned parsing for all the variables.

[0068] The list of tags and functions used in the template file are as follows:

[0069]

Tag	Function
TEMPLATE	Description of template
HEAD	Description of header
BODY	Description of main body
LINK	Description of related template files
VARIABLE	Declaration of template variable

INCLUDE Include of template file

SET Setting of variable value

EMBED Embedding of variable value

ISVALID Evaluation as to whether a variable value is valid or not

EVAL Comparison of variable values

LOOP Description of repetition

COMMENT Description of comment

<Use of cache value> A cache value comprises a variable name, a numerical value and a last update date combined in a set as shown in Fig. 9, and stored for each piece of information. An original object of using a cache value is to reduce the frequency of information acquisition based on SNMP and thus to achieve a shorter period of time before display of information. The last update date is compared with the date of information display request and the result is used as a reference for determination as to whether or not information is to be acquired by SNMP.

[0070] The flow of steps for use of a cache value upon failure of information acquisition by SNMP will now be described. Fig. 8 is a flowchart thereof.

[0071] When an information display request is presented to the Web server from the Web browser in step S801, the application is activated, and in step S802, information acquisition by SNMP is executed. When this information

acquisition is successful, the application embeds the acquired value into a parsed template file in step S804, and finally, sends it out as an HTML file to the browser. Further in step S805, if the cache file contains a cache value, it is written in as it is unless it is replaced by the acquired value.

[0072] On the other hand, if information acquisition is in failure, the cache value (S803) is written in the parsed template and displayed. At this point in time, the color or the character style of the character string to be displayed can be modified to indicate that the display shows, not the currently acquired value, but a previously stored cache value. When information acquisition by SNMP is not successful, the user can refer at least to a value acquired in the preceding run by using a value acquired successfully in the preceding run and updated as a cache value.

[0073]

[Other Embodiments] It is needless to mention that the object of the present invention can be achieved also by supplying a storage medium recording the program codes of the software implementing the functions of the above-mentioned embodiments to the system or the apparatus, and causing a computer (or CPU or MPU) of such a system or apparatus to read out and execute the program codes stored in the storage medium.

[0074] In this case, the program codes themselves read out from the storage medium implement the functions of the above-mentioned embodiments, and the storage medium storing these program codes would constitute the present invention.

[0075] Applicable storage media for supplying the program codes include, for example, a floppy disk, a hard disk, an optical disk, a magneto-optical disk, a CD-ROM, a CD-R, a magnetic tape, a nonvolatile memory card and a ROM.

[0076] By executing the program codes read out by the computer, the functions of the above-mentioned embodiments are implemented. In addition, this includes a case where the functions of the above-mentioned embodiments are implemented by causing the OS (Operating System) operating on the computer or the like to perform all or part of actual processing on the basis of the instructions of these program codes.

[0077] When the program codes read out from the storage medium are written in a memory provided in an expanded capability board inserted into the computer or in an expanded capability unit connected to the computer, and then, a CPU or the like provided in the expanded capability board or the expanded capability unit carries out all or part of actual processing in accordance with the instructions of the program codes, such a processing permits implementation of the functions of the above-mentioned embodiments. Such a

case is of course included here.

[0078] When applying the present invention to the above-mentioned storage medium, the program codes corresponding to the flowchart described above would be stored in this storage medium. Briefly, the program codes corresponding to the modules shown in Fig. 3 would be stored in the storage medium.

[0079]

[Advantages] According to the present invention, as described above, in failure of information acquisition by SNMP, it is possible for the user to refer at least to the value acquired in the preceding run by using the already acquired information from a successful information acquisition and stored as a cache value.

[0080]

[Brief Description of the Drawings]

[Fig. 1] Fig. 1 is a configuration diagram of LAN.

[Fig. 2] Fig. 2 is a configuration diagram of a PC on which the network management software is operable.

[Fig. 3] Fig. 3 is a module configuration diagram of the network management software.

[Fig. 4] Fig. 4 illustrates an example of the template file.

[Fig. 5] Fig. 5 illustrates a typical device list.

[Fig. 6] Fig. 6 illustrates another typical device list.

[Fig. 7] Fig. 7 is a file configuration diagram of the

network management software.

[Fig. 8] Fig. 8 is a flowchart regarding an information display method using the cache file of an embodiment of the present invention.

[Fig. 9] Fig. 9 illustrates an example of description of the cache file of an embodiment of the present invention.

[Reference Numerals]

- 100: Local area network (LAN)
- 101: Network board (NB)
- 102: Printer having open architecture
- 103: Personal computer (PC) connected to LAN 100
- 104: Personal computer (PC) connected to LAN 100
- 105: Printer connected to PC 104
- 106: File server connected to LAN 100
- 107: Network disk included in LAN 100
- 108: Print server included in LAN 100
- 109: Printer connected to print server 108
- 110: Local area network (LAN)
- 111: Personal computer (PC) connected to LAN 110
- 112: Personal computer (PC) connected to LAN 110
- 113: File server connected to LAN 110
- 114: Network disk included in LAN 110
- 115: Print server included in LAN 110
- 116: Printer connected to print server 110
- 120: Local area network (LAN)

121: Personal computer (PC) connected to LAN 120
122: Personal computer (PC) connected to LAN 120
130a: Modulation/demodulation (MODEM)/transponder
130b: Modulation/demodulation (MODEM)/transponder
140: Backbone
150: Personal computer (PC) connected to LAN 100, on which
www server operates
301: CPU executing network management program supplied by
FD212
302: ROM of PC 200
303: RAM of PC 200
304: System bus of PC 200
305: Keyboard controller (KBC) of PC 200
306: CRT controller (CRTC) of PC 200
307: Disk controller (DKC) of PC 200
308: Network interface card (NIC)
309: Keyboard (KB) of PC 200
310: CRT display (CRT) of PC 200
311: Hard disk (HD) of PC 200
312: Floppy disk drive (FD)
402: CGI interface
403: Overall control module
404: Parameter module
405: System module
406: System setting file

407: Device list module
408: Device search module
409: Device details module
410: Device intrinsic module
411: Protocol module
412: Template module
413: Template file
414: Cache file
901: WebNetSpot directory storing network management
software WebNetSpot
902: Document directory storing cache files
903: Images directory storing image files
904: Template directory storing template files
905: Network management software WebNetSpot
906: HTML file
907: Image file
908: SYS directory storing shared template files
909: PRODUCT directory storing template files dependent
upon device
910: NIC directory storing template files dependent upon
network board
911: Template file
912: Template file
913: Template file
1031: WWW browser program

1051: WWW server program
1052: Network management program
1061: WWW server program
1062: Network management software

FIG. 1

105: PRINTER
107: NETWORK DISK
109: PRINTER
102: PRINTER
104: PC (REMOTE PRINTER)
106: PC (FILE SERVER)
108: PC (PRINT SERVER)
150: PC (WWW SERVER)
103: PC (CLIENT)
130a: MODEM/TRANSPOUNDER
121: PC (CLIENT)
122: PC (CLIENT)
130b: MODEM/TRANSPOUNDER
140: BACKBONE
111: PC (CLIENT)
112: PC (CLIENT)
113: PC (FILE SERVER)
114: NETWORK DISK
115: PC (PRINT SERVER)
116: PRINTER

FIG. 2

304: SYSTEM BUS

FIG. 3

1061: WWW SERVER PROGRAM
402: CGI INTERFACE
403: OVERALL CONTROL
404: PARAMETERS
405: SYSTEM
406: SYSTEM SETTING FILE
407: DEVICE LIST
408: DEVICE SEARCH
410: DEVICE DETAILS
411: DEVICE INTRINSIC
412: TEMPLATE
413: TEMPLATE FILE
414: CACHE FILE
409: PROTOCOL

FIG. 4

TEMPLATE FILE STRUCTURE
DESCRIPTION OF HEAD BLOCK
DESCRIPTION OF BODY BLOCK
DESCRIPTION OF HEAD BLOCK
DESCRIPTION OF BODY BLOCK

FIG. 5

(1) FILE

- (2) EDITING
- (3) DISPLAY
- (4) JUMP
- (5) HELP
- (6) RETURN
- (7) NEXT
- (8) RE-READ
- (9) HOME
- (10) RETRIEVE
- (11) GUIDE
- (12) PRINT
- (13) SECURITY
- (14) STOP
- (15) BOOKMARK
- (16) PLACE
- (17) DEVICE DETAILS
- (18) PRODUCT NAME / MANAGER NAME / INSTALLATION PLACE /
EVALUATED SPACE / CONTACT
- (19) DEVICE LIST
- (20) STATE
- (21) COMMAND
- (22) SETTING
- (23) PRINTABLE
- (24) PAPER FEED TRAY
UPPER CASSETTE

LOWER CASSETTE

PAPER DECK

ENVELOPE FEEDER: EUROPEAN TYPE #4

RETURN TO TOP OF SECTION

(25) EQUIPMENT INFORMATION

(26) DEVICE DETAILS

[STATE-INFORMATION]

STATE

DEVICE

[NETWORK]

NETWORK BOARD

PROTOCOL

[ERROR LIST]

ERROR DETAILS

MANAGER MODE

(27) EMULATOR

(28) ... VACANT FLASH ROM CAPACITY

VACANT FLASH ROM CAPACITY

VACANT RAM CAPACITY

VACANT RAM CAPACITY

NUMBER OF CASSETTES

TWO-SIDE UNIT

ENVELOPE FEEDER

PAPER DECK
STAPLE STACKER
FONT
(29) YES
YES
YES
YES
(30) RETURN TO TOP OF SECTION
(31) RETURN TO TOP OF PAGE
(32) DEVICE INFORMATION
(33) DEVICE NAME
PRODUCT NAME
INSTALLATION PLACE
MANAGER NAME
CONTACT
(34) RETURN TO TOP OF SECTION
(35) RETURN TO TOP OF PAGE
(36) SETTING
(37) MR. EXPERIMENT
EVALUATED SPACE
(38) DOCUMENT END
(39) INTERNET
(40) RETRIEVE

FIG. 6

- (1) FILE
- (2) EDITING
- (3) DISPLAY
- (4) JUMP
- (5) HELP
- (6) RETURN
- (7) NEXT
- (8) RE-READ
- (9) HOME
- (10) RETRIEVE
- (11) GUIDE
- (12) PRINT
- (13) SECURITY
- (14) STOP
- (15) INTERNET
- (16) RETRIEVE

- (18) BOOKMARK
- (19) PLACE
- (20) DEVICE LIST
- (21) DEVICE DETAILS

[STATE · INFORMATION]

STATE

DEVICE
[NETWORK]
NETWORK BOARD
PROTOCOL
[ERROR LIST]
ERROR DETAILS
MANAGER MODE

(22) DEVICE DETAILS

(23) PRODUCT NAME / MANAGER NAME
INSTALLATION PLACE: EVALUATED SPACE / CONTACT

(24) NETWORK BOARD INFORMATION

(25) RESET

(26) INITIALIZATION

(27) BOARD NAME
FIRM VERSION
GANON-MIS VERSION
PHYSICAL INTERFACE
MAC ADDRESS
EFFECTIVE PROTOCOL
RETURN TO TOP OF SECTION

(28) PROTOCOL INFORMATION

(29) SETTING

(30) FRAME TYPE
NETWORK NO.
NODE ADDRESS

PRINT SERVER
PRINT SERVER NAME
TREE NAME
CONTEXT NAME
PACKET SIGN: NOT USED
PRINTER NO.
SERVICE MODE: MINIMUM PAPER CHANGE WITHIN QUEUE
FORM NO.
POLLING INTERVAL: 5 SEC
BUFFER SIZE
(31) RETURN TO TOP OF SECTION
(32) SETTING
(33) FRAME TYPE
IP ADDRESS ..
IP ADDRESS
SUBNET MASK
GATEWAY ADDRESS
BROADCAST ADDRESS
PRINT SERVICE
(34) RETURN TO TOP OF SECTION
(35) SETTING
(36) PHASE TYPE: PHASE 2
NETWORK NO.
NODE NO.
NAME

ZONE

TYPE

- (37) RETURN TO TOP OF SECTION
- (38) NUMBER OF RECEIVED PACKETS
- (39) DOCUMENT END

FIG. 8

(1) START

S801: INFORMATION ACQUISITION REQUEST FROM BROWSER

S802: IS INFORMATION ACQUIRED BY SNMP?

S803: USE CACHE VALUE

(2) SUCCESS

(3) FAILURE

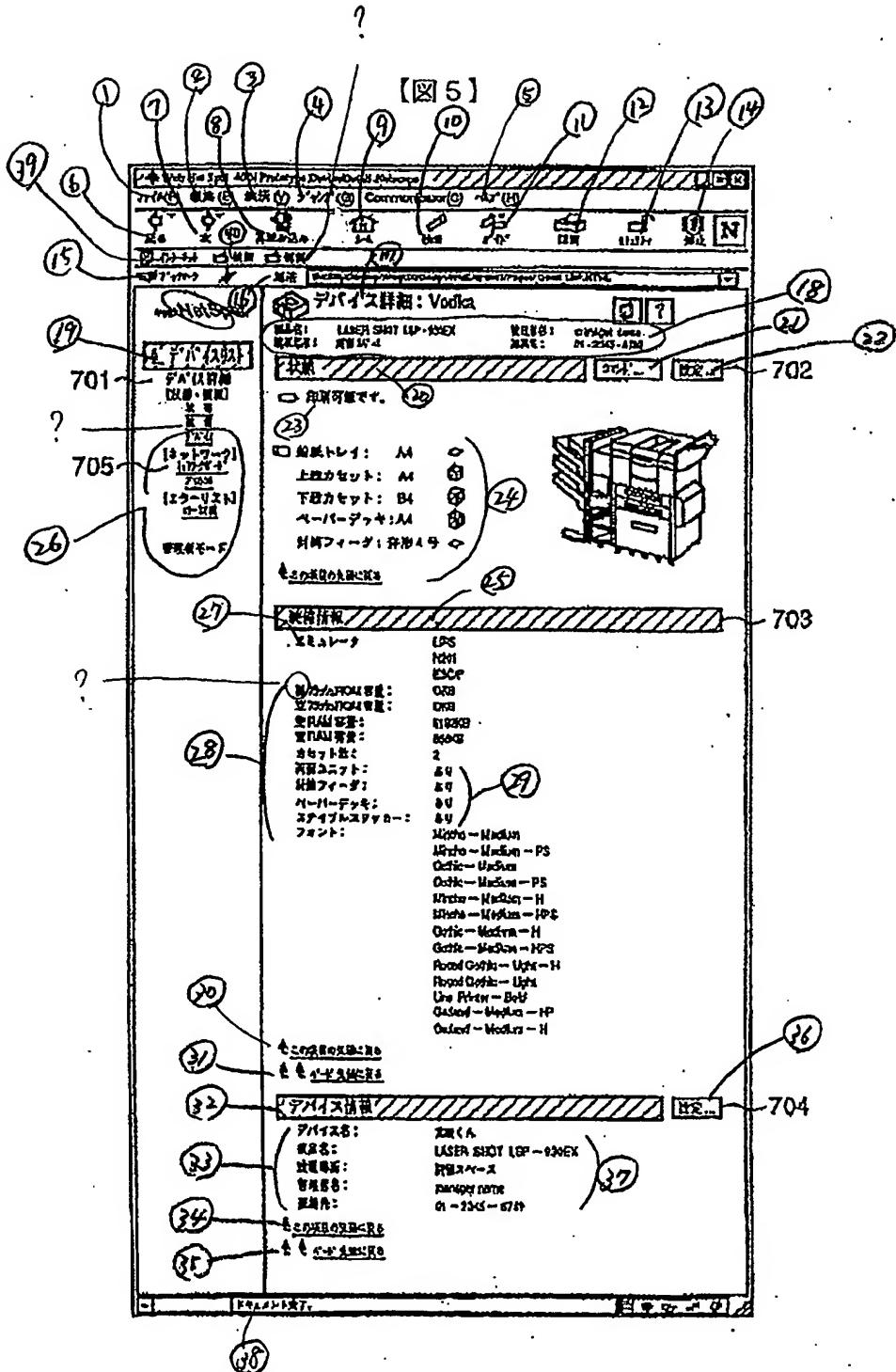
(4) END

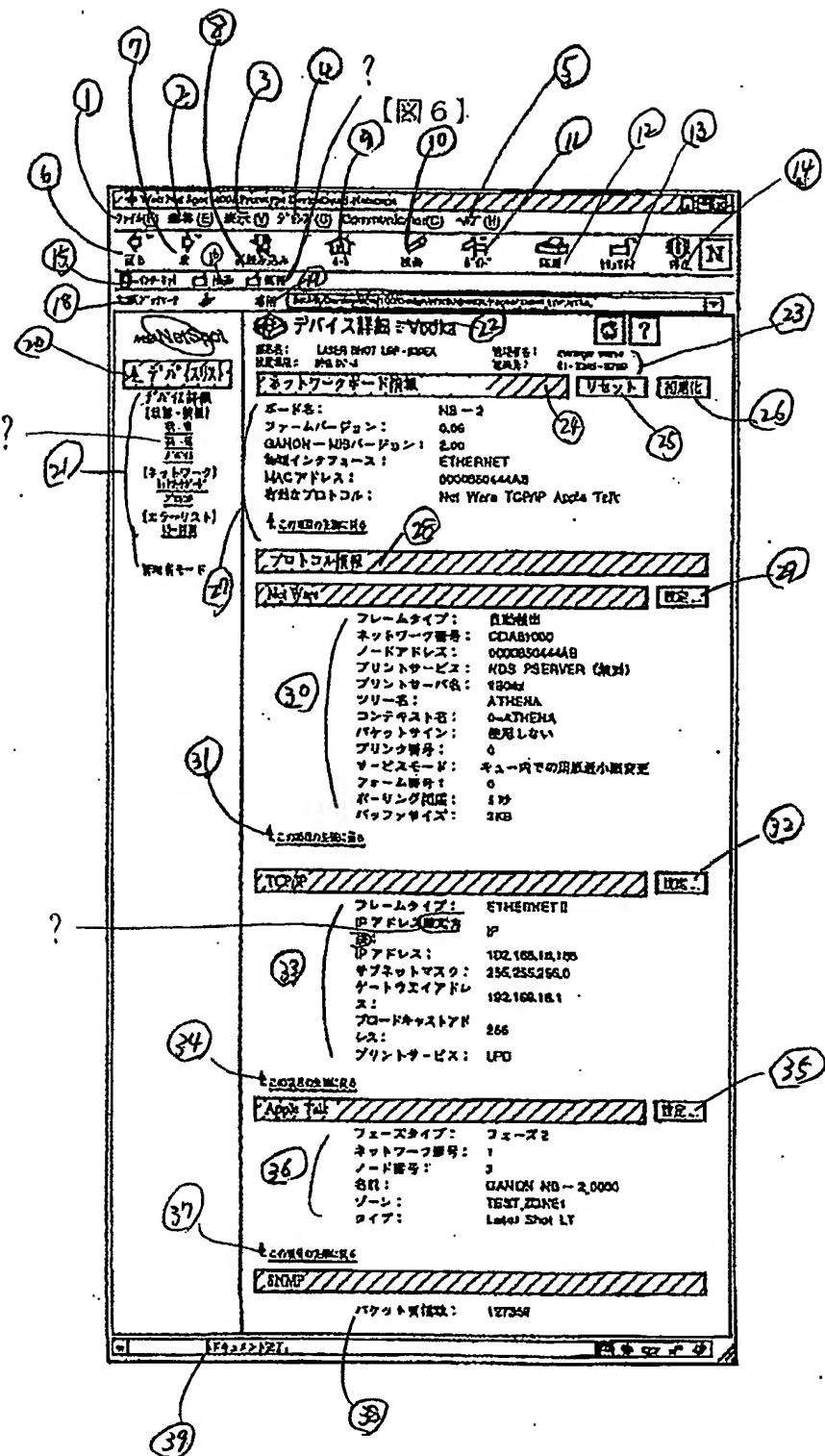
S804: USE ACQUIRED VALUE

S805: UPDATE CACHE VALUE

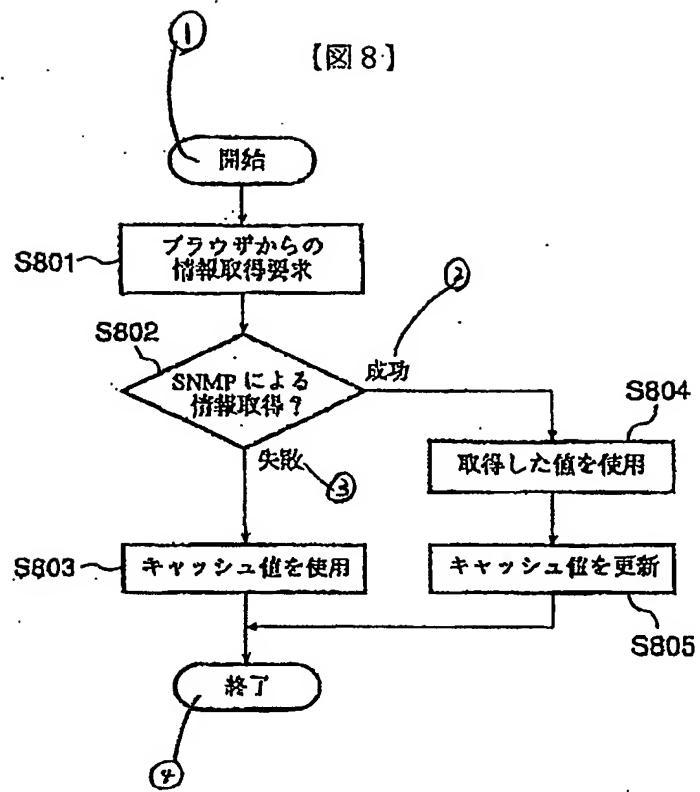
FIG. 9

A TYPICAL CACHE FILE





【図 8】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.